

FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

**Technisch-wissenschaftliches
Hochleistungsrechnen:
Herausforderungen komplexer Systeme
an die Computer-Simulation**

Christian Bischof, Friedel Hoßfeld*

FZJ-ZAM-IB-9821

Dezember 1998

(letzte Änderung: 18.12.98)

(*) RWTH Aachen

Preprint: Erscheint in „Die RWTH Aachen auf dem Weg ins 21. Jahrhundert“

Technisch-wissenschaftliches Hochleistungsrechnen: Herausforderungen komplexer Systeme an die Computer-Simulation

von

Christian Bischof und Friedel Hoßfeld

Zusammenfassung

Vor einem halben Jahrhundert präsentierte John von Neumann sein forschungsstrategisches Manifest über die Notwendigkeit von Hochleistungsrechnern und seine Entwurfsprinzipien des Digitalrechners. Seine Initiative war motiviert durch die Stagnation der analytischen mathematischen Methoden zur Lösung partieller Differentialgleichungen und getragen von der Idee, die Barriere der komplexen physikalischen und technischen Probleme durch Computersimulation mit neuen numerischen Verfahren zu überwinden.

Die Computersimulation hat sich seitdem mit den Fortschritten der Rechnerarchitektur und der Algorithmen zur dritten - Theorie und Experiment ergänzenden - Kategorie wissenschaftlichen Forschens entwickelt („virtuelles Labor“); sie ist gleichzeitig auf dem Wege, für die Industrie ein unverzichtbares Instrument zur Optimierung der Produktzyklen zu werden („virtuelles Produkt“).

Die wachsende Komplexität der Systeme und Prozesse in Forschung und Technik stellt gleichermaßen steigende Anforderungen an die Genauigkeit der mathematischen Modellbildung, an die Effizienz der numerischen (und nichtnumerischen) Methoden, an die Innovationskraft der Computerarchitektur und an neue Ansätze zur Interpretation und Interaktion mit großen Datenmengen. Die Ausnutzung von Parallelismus ist hierbei vonnöten, um die für zukünftige Problemlösungen nötige Rechenleistung zu liefern. Des weiteren ist die Visualisierung ein unverzichtbares Instrument (R.W. Hamming: „The Purpose of Computing is Insight, not Numbers!“). Diese inhärente Interdisziplinarität in der Anwendung verlangt aber auch neue Formen der Kooperation und neue Konzepte für die Ausbildung („Computational Science & Engineering“).

1. Wissenschaftliches Rechnen als Computational Science & Engineering

Es ist jetzt etwas mehr als 50 Jahre her, daß John von Neumann – gemeinsam mit seinem Kollegen Goldstine – sein Manifest über die Notwendigkeit der Entwicklung und die Entwurfsprinzipien des Digitalrechners verfaßte /1/. Sein Anstoß war die Stagnation der analytischen mathematischen Methoden zur Lösung partieller Differentialgleichungen, vornehmlich in der Strömungsdynamik, und er wollte mit seinem neuen Konzept des sequentiellen Digitalrechners, dessen Flexibilität seitdem den breiten Durchbruch des Computers in Wissenschaft, Wirtschaft und Gesellschaft bestimmt hat, den „digitalen Windkanal“ schaffen, um mit numerischen Methoden und der Simulation der komplexen Strömungsprozesse im Computer die Barriere der Stagnation zu durchbrechen.

Das Jahr 1946 ist somit das Geburtsjahr einer grundlegend neuen Methodik, die sich unter dem internationalen Namen „Computational Science & Engineering“ zur dritten, Theorie und Experiment ergänzenden Kategorie wissenschaftlichen Forschens entwickelt hat /2/. Ihre Methode ist die Simulation, ihr Instrument der Computer! Der wissenschaftliche Fokus von Computational Science & Engineering, deren strategische und inhaltliche Spannweite im Deutschen mit dem umfassenden Begriff „Wissenschaftliches Rechnen“ eingefangen wird, zielt auf komplexe Systeme, das heißt auf Probleme, die mit den analytischen Methoden allein einer Lösung nicht näher gebracht werden können.

Es liegt in der Natur der Sache, daß diese Problemstellungen neben effizienten Algorithmen, Software-Werkzeugen und Programmiermodellen leistungsfähige Computerarchitekturen bis hin zu Supercomputern erfordern. Das Rechenzentrum der RWTH Aachen betreibt z. B. im Auftrag des Landes Nordrhein-Westfalen den sogenannten „Landesvektorrechner“, eine Fujitsu VPP300/8 (8 CPUs) während das Forschungszentrum Jülich als Supercomputing-Zentrum des Bundes einen Supercomputer-Komplex aus CRAY T3E (512+256 Prozessoren), T90 (16 CPUs) und J90 (16+12 CPUs) betreibt.

2. John von Neumanns Idee des „Digital Windtunnel“

Die Entwicklung der Mathematik von Differentialgleichungen – und Variationsmethoden – begann mit der Entwicklung der neueren Astronomie und Physik und war von Anbeginn diesen Wissenschaften aufs engste verbunden. Entsprechend der Vielfalt und Schwierigkeit der behandelten Aufgaben haben sich die großen Mathematiker und Physiker des 18. und 19. Jahrhunderts zunächst wissenschaftlich interessanten und auch technisch brennenden Einzelfragen zugewandt. Sie gelangten zwar noch nicht zu einer geschlossenen Theorie der partiellen Differentialgleichungen, doch spiegeln selbst „kleine“ Darstellungen /3,4/ und großartige Lehrbücher der theoretischen Physik /5/ die Schönheit und Harmonie in den Differentialgleichungen der Physik.

In einem denkwürdigen Vortrag über „High-Speed Computing Devices and Mathematical Analysis“ beim First Canadian Mathematical Congress im Juni 1945 identifizierte John von Neumann erstmalig die numerische Hydrodynamik, auszuführen auf elektronischen Digitalrechnern, als ein Schwerpunktgebiet zukünftiger Forschung. Diese Vorstellungen präsentierte er mit seinem epochemachenden Positionspapier „On the Principles of Large Scale Computing Machines“ im Mai 1946 dem Mathematical Computing Advisory Panel des Office of Research and Inventions im Navy Department in Washington /1/. Damals faßte er die Situation der theoretischen Hydrodynamik so zusammen: „The advance of analysis is, at this

moment, stagnant along the entire front of nonlinear problems. ... Mathematicians had nearly exhausted analytic methods which apply mainly to linear differential equations and special geometries“.

John von Neumann wollte den Digital Windtunnel. Er schlug vor, daß man die analytischen durch numerische Methoden ersetzen und die Entwicklung der Digitalrechner und ihre Nutzung fördern solle, da digitale Maschinen viel schneller gemacht und mit höherer Flexibilität und Genauigkeit ausgestattet werden könnten als Analogrechner, zu denen er auch die damals für Strömungsexperimente weithin gebrauchten Windkanäle zählte.

Unmittelbar nach seinem Manifest legte John von Neumann gemeinsam mit A. W. Burks und H. H. Goldstine in mehreren Berichten den logischen Entwurf der nach ihm benannten Rechnerarchitektur vor /6-9/. Damit war das neue Konzept der digitalen Maschine definiert, das den Durchbruch des Computers schaffte.

John von Neumann hat entscheidend dazu beigetragen, daß die konzeptionelle und methodische Unsicherheit in den Jahrzehnten nach 1946 allmählich aufgehoben wurde und auch eine gewisse Aussöhnung zwischen Experiment und mathematischer Analysis zustande kam. Die numerische Lösungsmethodik für die zugrundeliegenden nichtlinearen Gleichungen führte – über die Diskretisierung von Raum und gegebenenfalls Zeit – zwingend in die lineare Algebra und ihre der Numerik erschließbaren Konzepte und Algorithmen /10-12/.

Die Computersimulation entwickelte sich in der Tat zunehmend zu der nützlichen Quelle der Einsicht in die Prozesse der Strömungsmechanik, wie sie sich von Neumann erhofft hatte; ja, Computational Science & Engineering wuchs sich zur dritten Säule wissenschaftlichen Forschens aus. Es mag die – wenn auch spät – weltweit in Gang kommende Annäherung der Informatik an Computational Science & Engineering verdeutlichen, daß die Association of Computing Machinery (ACM) bei ihrem 50th Anniversary Symposium: Perspectives in Computer Science auf das Zukunftspotential der CFD im Verbund mit den neuen Generationen von – parallelen – Supercomputern eingegangen ist, indem sie die Dringlichkeit der numerischen Simulation bei der Lösung der komplexen dreidimensionalen Probleme bei reaktiven Strömungen herausgestellt hat /13/.

Hieraus wird ersichtlich, daß trotz der unzweifelhaft großen Fortschritte in der numerischen Lösung partieller Differentialgleichungen, wie sie etwa in den Monographien von W. Hackbusch über elliptische Differentialgleichungen und über Mehrgitterverfahren /14,15/ und S. F. McCormick über adaptive Multilevel-Methoden /16/ ihren Niederschlag gefunden haben, und trotz der dramatisch gestiegenen Leistung der Supercomputer /17,18/ die numerische Simulation von komplexen Problemen eine permanente Herausforderung an die Wissenschaft darstellt.

3. Antworten der Rechnerarchitektur

Die Lösung mehrdimensionaler unstationärer Differentialgleichungen stellte schon immer höchste Herausforderungen sowohl an die numerischen Methoden wie auch die Rechnerarchitekturen.

Vektorrechner. Die erste Antwort der Rechnerarchitektur auf diese Herausforderungen der partiellen Differentialgleichungen waren die Vektorrechner, die mit ihrem Pipeline-Prinzip die

sequentiell überlappende arithmetische Verarbeitung von Datenstrukturen der linearen Algebra, der „Vektoren“, optimierten. Aber schon 1982 machte Cray Research den Schritt zu Mehrprozessor-Vektorrechnern mit Multitasking und damit in die Parallelverarbeitung /19/. Damit ging auch die rechtzeitige Entwicklung zu offenen Systemen einher, wie sie sich in Unix-basierten Betriebssystemen und der Unterstützung der TCP/IP-Kommunikation manifestiert – zwei innovative Strömungen, die gleichfalls aus der Welt der Wissenschaft und Forschung kommend die Entwicklung des Computing auf einer breiten Wellenfront, von den Workstations bis zu den Netzwerken, in die Zukunft getragen haben. Die heutigen Vektorrechner der obersten Leistungsklasse sind vielfach noch die „Arbeitspferde“ für Computational Science & Engineering, wenngleich sich mehr und mehr die neuen super-skalaren Multicomputer die Arbeit mit ihnen teilen /19/.

Multicomputer. Der diesen Maschinen zugrundeliegende Denkansatz ist, auf die Entwicklung von speziellen Prozessoren zu verzichten und sich stattdessen auf die Integration von Prozessoren zu konzentrieren, die auch in PCs und Workstations eingesetzt werden, um auf diese Weise von dem Preisdruck auf dem sehr viel größeren Markt für diese Bauteile zu profitieren. Die hier zugrundeliegenden Prozessoren sind in der Lage, in einem Zeittakt mehrere verschiedene Gleitpunktoperationen (z.B. eine Addition und eine Multiplikation) auszuführen, und werden deshalb als „superskalar“ bezeichnet. Der Haufen PCs („pile of PCs“) als Supercomputer /20/ ist sicher ein attraktiver Gedanke und für ganz bestimmte Klassen von Anwendungen wurden auf billig vernetzten Ensembles von PCs auch erstaunliche Leistungen erzielt /21/. Neben dem einzelnen Prozessor spielt aber auch die Kommunikation zwischen Prozessoren, die Anbindung von großen Speichern und externen Medien eine große Rolle bei der Nutzbarkeit eines Computersystems, insbesondere wenn die Nutzung im Rahmen eines breiten Anwendungsspektrums vorgesehen ist. Diese Systemaspekte werden nicht vom Massenmarkt getragen, stellen Computerintegratoren vor hohe technische Anforderungen und sind im Höchstleistungsbereich dementsprechend teuer, egal welcher Prozessortyp dem System zugrundeliegt.

Heterogene Systeme. Die effiziente Nutzung von Höchstleistungssystemen setzt eine enge Abstimmung zwischen Rechnerstruktur und zugrundeliegendem Algorithmus voraus. Es ist unmittelbar einsichtig, daß die unterschiedlichen Datenflußstrukturen der Algorithmen (z.B. der Fast Fourier Transform und der Mehrgittermethoden) sich nicht gleichermaßen gut auf ein fest vorgegebenes Verbindungsnetzwerk zwischen den Prozessoren abbilden lassen. Hinzu tritt das Problem der gleichmäßigen Lastverteilung. Große Programme setzen sich in der Regel aus sehr verschiedenen Algorithmen zusammen; sie sind algorithmisch heterogen. Die Erfahrungen mit den verschiedenen Supercomputer-Architekturen und ihren Stärken und Schwächen, die technologischen Hindernisse für größere Leistungssprünge bei der Vektorverarbeitung, die große Schwankungsbreite in den Leistungsdaten für Algorithmen auf verschiedenen Parallelrechnern führen ganz natürlich zur Idee des heterogenen Rechnens und der heterogenen Computer /22/. Auf heterogenen Systemen wird die Arbeit auf unterschiedliche Rechner so verteilt, daß die charakteristische Einzelaufgabe, sei sie skalar, vektoriell, vektor-parallel oder massiv-parallel, jeweils auf derjenigen Rechnerarchitektur abgearbeitet wird, die nachweislich am besten für diesen Aufgabentypus geeignet ist. Hierdurch wird sowohl die Effizienz des Rechnens als auch das Kosten/Nutzungsverhältnis der beteiligten Systeme erhöht.

4. Herausforderungen an Programmierwerkzeuge

Obwohl viele aktuelle technisch-wissenschaftliche Fragestellungen von einer fast überwältigenden Komplexität sind, konnten in den letzten Jahren zunächst konzeptionell, dann auch durch Implementierungen wirkungsvolle Ideen umgesetzt werden. Der Entwurf paralleler Algorithmen sowie die effiziente Umsetzung auf reale Systeme bleibt aber die große Herausforderung an die Mathematik und Informatik. Die Schwierigkeiten gründen in den durch Kommunikation, Parallelismus und Speicherhierarchien bedingten engen Wechselbeziehungen zwischen einem Algorithmus und seiner Abbildung auf ein Computersystem.

Diese Problematik wird vielleicht am deutlichsten, wenn man sich vor Augen hält, daß schon auf nur einem Prozessor, wie er bei High-End-Workstations eingesetzt wird, die Leistung bei einem so einfachen Algorithmus wie der Matrix-Matrix Multiplikation sich um einen Faktor über 100 unterscheiden kann, je nachdem ob man den Textbuchalgorithmus ohne aggressiven Gebrauch von Compiler-Optionen compiliert oder auf eine auf die Architektur abgestimmte Assemblerroutine zurückgreift. Das Zusammenschalten von mehreren solcher Knoten in einem parallelen System erhöht die Anzahl der Programmierungsfreiheitsgrade weiter.

Da der Programmieraufwand für komplexe Fragestellungen die Produktzyklen bei der Hardware von zwei bis vier Jahren im allgemeinen überschreitet, kommt Programmierwerkzeugen eine herausragende Rolle zu. Sie dienen nicht nur dazu, um die Programmierung zu erleichtern, sondern auch um über verschiedene Hardware-Generationen hinaus den Benutzern eine wohldefinierte Programmierabstraktion in die Hand zu geben und so die Investition in parallele Software langfristig attraktiv zu gestalten.

Das Programmiermodell des Message Passing wird auf Parallelrechnern auf breiter Basis effektiv genutzt. Mit dem MPI-Standard /23/ wurde eine Programmierschnittstelle definiert, welche von allen Herstellern paralleler Systeme unterstützt wird. Das MPI-Modell geht davon aus, daß jeder Prozeß direkten Zugriff auf einen eigenen Speicherbereich hat und auf andere Daten durch expliziten Austausch von Botschaften mit anderen Prozessen zugreift. Dieses Programmiermodell ist effizient umsetzbar und transparent; für den darin ungeübten Benutzer ist es aber nicht unbedingt einfach, korrekte und effiziente Programme zu erstellen.

Ein alternatives Programmiermodell ist Shared Virtual Memory (SVM) (siehe z.B. /24/), ein Konzept, das dem Benutzer erlaubt, auf alle Daten wie in einem traditionellen globalen Adreßraum zuzugreifen. Dieses Konzept ist komfortabel, aber eine effiziente Umsetzung eines Benutzerprogramms auf parallele Hardware durch Compiler kann nur dann stattfinden, wenn die vom Benutzerprogramm induzierten Datenzugriffe gewisse Lokalitäts- und Regelmäßigkeitsvoraussetzungen erfüllen. Selbst dann ist dies heute eine nichttriviale Aufgabe, die schon bei dichtbesetzten Felddatenstrukturen wie z.B. in HPF /25/ höchste Anforderungen an Compiler stellt /26/, da sie in vielen Fällen die Abbildung von SVM auf ein MPI-basiertes Programmiermodell beinhaltet.

In beiden Fällen kommt Werkzeugen zum Verständnis des Programmverhaltens herausragende Bedeutung zu. Zum einen benötigt man leistungsfähige Debugger, um den Benutzer dabei zu unterstützen, Fehler in den (z.T. asynchron laufenden) Programmen zu finden. Zum anderen sind Systeme zum Verständnis der Wechselwirkung zwischen Programm und zugrundeliegender Architektur von großer Bedeutung, da insbesondere bei großen parallelen Konfigurationen enorme Leistungsdifferenzen zwischen verschiedenen Implementationen derselben Problemlösung auftreten können. Diese Werkzeuge spielen

mithin eine große Rolle beim Verständnis der Skalierbarkeit von Programmen. Ein Beispiel für ein Software-Werkzeug, das die Leistungsanalyse und die Fehlerdiagnose unterstützt, ist das Analyse- und Visualisierungswerkzeug VAMPIR /24/, das ursprünglich vom Forschungszentrum Jülich entwickelt wurde, jetzt an der Technischen Universität Dresden (Prof. Dr. W. E. Nagel) weiterentwickelt wird und mittlerweile von der Pallas GmbH vermarktet wird.

5. Visualisierung und Virtual Reality

Die Visualisierung ist eine wesentliche konzeptionelle Grundlage für wissenschaftliche und technische Erkenntnisse aus der Computersimulation komplexer System- und Prozeßzusammenhänge. Sie ist vor allem für die Behandlung und das Verständnis komplizierter dynamischer Vorgänge, deren wissenschaftlich-technische Analyse die mehrdimensionale Repräsentation der Prozesse in Raum und Zeit erfordert, unabdingbar. Die neuen Konzepte der Modellbildung in der Wissenschaft und der Produktentwicklung in der Industrie, „Virtuelles Labor“ und „Virtuelles Produkt“, gründen auf dem Potential der neuen leistungsfähigen Visualisierungs- und Multimediatechniken /27/.

Für eine menschengerechte, d.h. anschauliche Repräsentation visueller, aber insbesondere auch nicht-visueller, simulierter Phänomene spielen computergraphische Methoden eine herausragende Rolle. Die wachsende Leistungsfähigkeit der Computerhardware ermöglicht hier nicht nur eine hohe Komplexität der Simulation selbst, sondern wurde und wird ebenfalls zur Entwicklung immer komplexerer und effizienterer Visualisierungstechniken genutzt. Vektorisierung und Parallelisierung zur schnellen visuellen Abbildung mehrdimensionaler Daten haben auch in der Computergraphik längst Einzug gehalten. Hat man sich in der Vergangenheit aufgrund der unzureichenden Leistung verfügbarer Graphikcomputer mit statischen Darstellungen oder bestenfalls Animationen zufrieden geben müssen, rückt heute die interaktive Exploration selbst komplexer, dynamischer Szenarien in den Bereich des Machbaren /28/. Ein interaktives Eingreifen in Simulationen, welche nur auf Hochleistungsvektor- und Parallelcomputern rechenbar sind, wird über eine schnelle Vernetzung von Visualisierungs- und Simulationscomputer möglich.

Durch das Hinzufügen akustischer und ggf. haptischer Informationen wandelt sich die Disziplin der wissenschaftlichen Visualisierung in Richtung einer multimedialen Erfahrung von im Computer simulierten, physikalischen oder abstrakten Phänomenen /29-30/. In diesem Kontext ist auch die Erweiterung der traditionell zweidimensional ausgelegten Schnittstelle zwischen Mensch und Computer auf drei Dimensionen zu sehen. Insbesondere stereoskopische Visualisierungstechniken erlauben im Vergleich zu lediglich perspektivischen Darstellungen ein wesentlich intuitiveres Verständnis räumlicher Zusammenhänge. In technisch-wissenschaftlichen Anwendungsfeldern haben sich Systeme durchgesetzt, welche auf Großbild-Stereoprojektionen basieren. In der sogenannten CAVE /31/ werden mehrere Projektionsflächen zu einem virtuellen Raum zusammengesetzt. Für viele Anwendungen des wissenschaftlichen Visualisierens reichen platzsparende und kostengünstige Systeme aus, die über lediglich eine oder maximal zwei Abbildungsflächen verfügen /30/. Ordnet man eine Projektionsfläche horizontal an, erhält man einen Tisch, auf dem virtuelle Objekte manuell bearbeitet und simulierte Prozesse von allen Seiten betrachtet werden können /32/.

Die Vernetzung mehrerer Projektionstische erlaubt eine kooperative Exploration und Diskussion von Simulationsergebnissen über räumliche Entfernungen hinweg /33/. Im

Gegensatz zu CAVEs, welche aufgrund ihrer mehrfachen Projektionsflächen bis dato teure Spezialhardware erfordern, können Projektionstische bereits mit der neuesten Generation von PCs betrieben werden. Es ist deshalb zu erwarten, daß Projektionstische in den nächsten Jahren eine große Verbreitung finden und zum alltäglichen Werkzeug des Wissenschaftlers werden.

6. Anwendungsbeispiel: Simulation einer Gasturbine

Die Entwicklung von effizienten Algorithmen und zuverlässigen Programmen ist heute an einem Punkt angekommen, an dem dreidimensionale Navier-Stokes-Gleichungen zur Strömungsberechnung um komplexe Geometrien wie beispielsweise Flugzeuge oder Turbinenbeschaufelungen numerisch gelöst werden können. Zum Beispiel wurde am Institut für Dampf- und Gasturbinen der RWTH Aachen (Prof. Dr.-Ing. D. Bohn) das Programm CHTFlow entwickelt, welches die dreidimensionalen Navier-Stokes-Gleichungen basierend auf einem Ansatz der finiten Volumen löst. Darin wird ein implizites Zeitschrittverfahren angewendet, wobei die Lösungen der auftretenden linearen Gleichungssysteme mit einem iterativen Gauß-Seidel-Verfahren berechnet werden /34/. Damit lassen sich realistische Simulationen z. B. einer vierstufigen Turbine durchführen, wie sie in Bild 1 dargestellt ist.

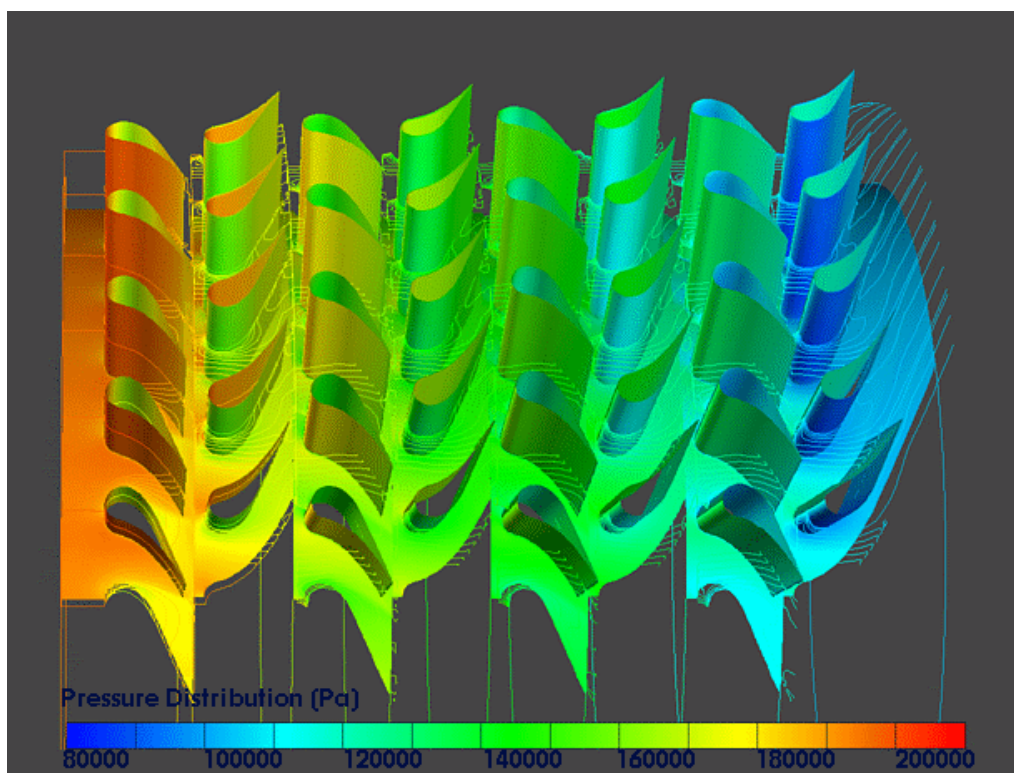


Bild 1

Das Bild zeigt die Druckverteilung auf den Profil- und Nabenoberflächen sowie Isobaren im Kanalmittelschnitt der vierstufigen Axialturbine. Auf dem Umfang der Nabe sind fünf der 32 Leitschaufeln und fünf der 33 Laufschaufeln aller vier Stufen dargestellt. Der fehlende Teil der Nabe ist durch Linien angedeutet, die entsprechend dem Druckniveau gefärbt sind. Die Visualisierung der Ergebnisse wurde am Rechenzentrum der RWTH Aachen, die Berechnung auf dem Vektorrechner CRAY J90 am Zentralinstitut für Angewandte Mathematik am

Forschungszentrum Jülich durchgeführt. Das hier gezeigte Problem umfaßt 650 000 Gitterpunkte, unter der Annahme, daß die Strömung von Schaufelkanal zu Schaufelkanal periodisch ist.

Sinn und Zweck einer Turbine ist es, potentielle Strömungsenergie in Form von Druckenergie in mechanische Energie zu verwandeln, die über die rotierende Welle in der Turbine übertragen wird, z.B. um einen Verdichter oder einen Generator anzutreiben. Eine „Stufe“ in einer Turbine besteht aus einer stehenden Schaufelreihe, dem Leitrad und einer rotierenden Schaufelreihe, dem Laufrad. Das Leitrad bewirkt eine Umlenkung der Strömung in Umfangsrichtung, so daß die Strömung mit einem Drall in das nachfolgende Laufrad gelangt. Die Kräfte des Dralles, die auf die Laufschaufeln wirken, dienen zum Antrieb der rotierenden Welle. Aus diesem thermodynamischen Prozeß ergibt sich eine Absenkung des Druckes und der Temperatur.

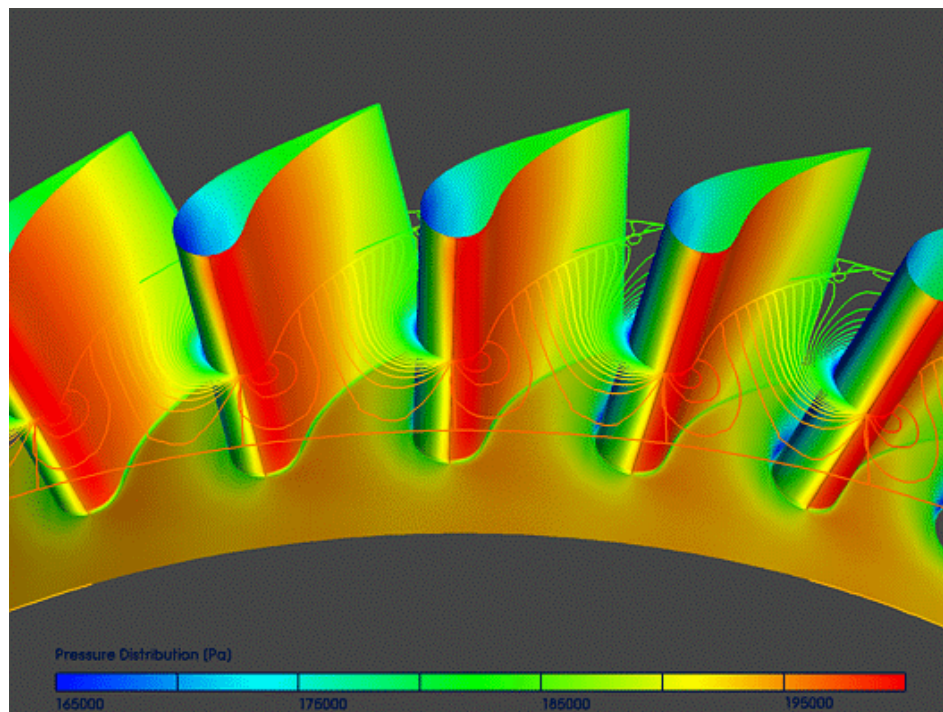


Bild 2

Im Bild 2 ist lediglich die Druckverteilung in der ersten Leitschaufelreihe der Turbine dargestellt. Dadurch daß die Druckverteilung auf einem kleineren Bereich (1,65-2,1 bar) dargestellt ist, sind die Gradienten innerhalb des Leitrades besser zu erkennen. Der Bereich der Schaufel (rot), der mit einem hohen Druck beaufschlagt ist, wird als Druckseite bezeichnet, und der Bereich der Schaufel (blau), der mit einem geringen Druck beaufschlagt ist, wird als Saugseite bezeichnet. In den Darstellungen hier sind die Schaufeln hohl visualisiert, so daß in das Innere der Schaufeln gesehen werden kann.

7. Schlußfolgerungen

Die aus dem Zusammenwirken von Modellbildung, Computersimulation und Visualisierung resultierende Verbindung aus mathematischer Methodik und informationstechnischem Instru-

mentarium hat die strategische Disziplin des Wissenschaftlichen Rechnens zu einer die internationale Wettbewerbsfähigkeit von Wissenschaft und Wirtschaft mitbestimmenden Schlüsseltechnologie gemacht. Die Stärkung dieser Technologie erfordert jedoch nicht nur Fortschritte in der Suche nach besseren Algorithmen, Programmiermethodiken, Rechnerarchitekturen und Visualisierungsansätzen, sondern aufgrund ihres interdisziplinären Charakters die interfakultative Aufgeschlossenheit für die wissenschaftlichen Fragen der anderen Fachgebiete, da ohne ein solches fächerübergreifendes Denken die Komplexität der zu bewältigenden Probleme nicht bewältigt werden kann. Zu diesem Zweck benötigt man wohl auch hierzulande neue Curricula, wie sie in den USA und auch in anderen Ländern Europas für Computational Science & Engineering an vielen renommierten Hochschulen bereits eingeführt worden sind /2/.

Acknowledgements. C.B. dankt Herrn Dr. T. Kuhlen vom Rechenzentrum der RWTH Aachen für seine freundliche Unterstützung in Fragen der Virtuellen Realität und Herrn Dipl.-Ing. W. Risch vom Rechenzentrum der RWTH Aachen für das Überlassen der Ergebnisse der Turbinensimulation.

Literatur

- /1/ von Neumann, J., and Goldstine, H.H., in: Collected Works Vol. V, p. 1-32.
- /2/ Sameh, A., and Riganati, J., IEEE Computer 26(1993), No. 10, 8-12.
- /3/ Sauter, F., Differentialgleichungen der Physik, Sammlung Götschen Bd. 1070, Berlin, 1958.
- /4/ Gröbner, W., und Lesky, P., Mathematische Methoden der Physik II, BI Hochschultaschenbücher 90/90a, Mannheim, 1965.
- /5/ Sommerfeld, A., Vorlesungen über theoretische Physik, Bd. VI: Partielle Differentialgleichungen der Physik, 4. Auflage, Akademische Verlagsgesellschaft Geest & Portig, Leipzig, 1958
- /6/ Burks, A.W., Goldstine, H.H., and von Neumann, J., Collected Works, Vol. V, p. 34-79
- /7/ Goldstine, H.H., and von Neumann, J., Collected Works, Vol. V, p. 80-151
- /8/ Goldstine, H.H., and von Neumann, J., Collected Works, Vol. V, p. 152-214
- /9/ Goldstine, H.H., and von Neumann, J., Collected Works, Vol. V, p. 215-253
- /10/ Wilkinson, J.H., J. ACM 18(1971), 137.
- /11/ Ortega, J.M., Introduction to Parallel and Vector Solution of Linear Systems, Plenum Press, New York, 1988.
- /12/ Ortega, J.M., Voigt, R.G., and Charles Romine, A Bibliography on Parallel and Vector Numerical Algorithms, in Parallel Algorithms for Matrix Computations, K. A. Gallivan et al., p.125-197, 1990.
- /13/ Cinnella, P., ACM Computing Surveys 28(1996), No. 1, 93-96.
- /14/ Hackbusch, W., Theorie und Numerik elliptischer Differentialgleichungen, Teubner Studienbücher - Mathematik, B. G. Teubner, Stuttgart, 1986.
- /15/ Hackbusch, W., Multi-Grid Methods and Applications, Springer, Berlin, 1985.
- /16/ McCormick, S.F., Multilevel Adaptive Methods for Partial Differential Equations, Frontiers in Applied Mathematics Vol. 6, SIAM, Philadelphia, 1989.
- /17/ Kuwahara, K., Intern. J. High Speed Computing 4(1992), No. 1, 49-70.
- /18/ Top500 Benchmark. www.top500.org.
- /19/ Hwang, K., Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill, New York, 1993

- /20/ Hill, J., Warren M., and M. P. Goda, M.P., I'm not going to pay a lot for this supercomputer!. Linux Journal, 45, 1998. <http://www.ssc.com/lj/issue45/2392.html>.
- /21/ SterlingT. , Cwik, T. , Becker, D., Salmon, J., Warren, M., and Nitzberg, B., An assessment of Beowulf-class computing for NASA requirements: Initial findings from the first NASA workshop on Beowulf-class clustered computing. In Proceedings, IEEE Aerospace Conference. March 21-28, Aspen CO, 1998. Online at http://loki-www.lanl.gov/papers/ieee_aero98/p312.ps
- /22/ Khokhar, A. A., et al., IEEE Computer 26(1993), No. 6, 18-27
- /23/ Gropp, W., and Lusk, E., and Skjellum, A. Using MPI -- Portable Parallel Programming with the Message Passing Interface, MIT Press, Cambridge, 1994.
- /24/ Hossfeld, F., and Nagel, W.E., Per Aspera ad Astra: On the Way to Parallel Processing, in: H.W. Meuer (Hrsg.): Supercomputer 1995, Anwendungen, Architekturen, Trends, FOKUS - Praxis Information und Kommunikation Bd. 13, K. G. Saur, München, 1995, S. 246-259
- /25/ Koelbel, C., Loveman, D., Schreiber, R., Steele, G. Jr., Zosel, M., The High Performance Fortran Handbook, MIT Press, Cambridge, 1994.
- /26/ Haghighat, M., Symbolic Analysis for Parallelizing Compilers, Kluwer Academic Publishers, Dordrecht, 1995.
- /27/ Woodward, P.R., IEEE Computer 26(1993), No. 10, 13-25
- /28/ Haase, H., Dai, F., Strassner, J., and Göbel, M.: Immersive Investigation of Scientific Data, In: Hagen, H., Nielson, G., and Müller, H. (eds.), Scientific Visualization - Overview, Methodologies and Techniques, IEEE Computer Society Press, 1997
- /29/ Eckel, G., Göbel, M., Hasenbrink, F., Heiden, W. Lechner, U., Tramberend, H., Wesche, G., and Wind, J.: Benches and Caves, In: Proc. of 24th Annual Conference of the IEEE Industrial Electronics Society (IECON'98), 1998.
- /30/ Steffan, R., and Kuhlen, T.: A Virtual Workspace Including a Multimodal Human Computer Interface for Interactive Assembly Planning, In: IEEE International Conference on Intelligent Engineering Systems (INES'98), 1998.
- /31/ Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R.V., and Hart, J.C.: The CAVE: Audio Visual Experience Automatic Virtual Environment, Communications of the ACM, 35(6), 1992.
- /32/ Krüger, W., Bohn, C.A., Fröhlich, B., Schüth, H., Strauss, W., and Wesche, G.: The Responsive Workbench, a Virtual Work Environment, IEEE Computer 28(7), 1995.
- /33/ Disz, T.L., Papka, M.E., Pellegrino, M., and Stevens, R.: Sharing Visualization Experience among Remote Virtual Environments, In: Proc. of the International Workshop on High Performance Computing for Computer Graphics and Visualization, Swansea, UK, 1995.
- /34/ R. Emunds, I. Jennions, D. Bohn, J. Gier: The Computation of Adjacent Blade-Row Effects in a 1.5 Stage Axial Flow Turbine, Paper 97-GT-81, ASME Turbo Expo '97, Orlando, USA, June 1997